This document contains the original review of the NWIS XML format but Ilex Engineering and also contains the NWIS response to each recommendation. The NWIS responses are in this color blue and are bold and in a larger font.

1 Introduction

The purpose of this document is to analyze and review the XML format design specified by “hydroml.dtd”, which can be downloaded from:

`http://water.usgs.gov/nwis_activities/xml/nwis_hml.html`

The analysis was done by diagramming the XML entities as classes using Unified Modeling Language. The approach taken for the class diagrams is explained in section 1.1.

In this document, we have identified some issues which may require further consideration by USGS. These are presented in the narrative as questions, and are shown in *italics.* Recommendations for modifications to the design are shown in **bold face.**

1.1 UML Diagrams Representing XML DTD

In this document, The HYDROML DTD is represented using a series of UML class diagrams. The following DTD to UML mapping is used:

- Entities are shown as classes.
- Attributes are shown as data within the class box, along with the type (usually CDATA). Required attributes are underlined.
- Attributes that have enumerated types are shown with type "enum" and, where practical, a comment box listing the values.
- Attributes that have the enumerated types "Yes" and "No" are shown with type "Boolean".
- Content is shown using the XML aggregation relationship. Cardinality of the relationship is always shown.
- If an entity contains #PCDATA content, "#PCDATA (content)" is shown within the class box as the final attribute.
- In the case of the (many) Measurement-Value structures, a stereotype "MeasurementValue" is defined and appears at the top of the class box.
- In the few cases where a class contains one of many possible types, this is shown as inheritance.

2 Hydroml XML Strategy

2.1 Hydroml DTD

The HydroML DTD contains excellent documentation on individual entites, attribute formats and content. The only thing lacking is a top-level description of the use-cases in which HydroML files will be used, and the varying content expected in each use case.

For example, Computations and sub-entities will not be included in field-generated files. Site-Visit information may or may not be included in rating-calculation files shared with other agencies.

Recommend: Additional documentation at top-level of the DTD. This should describe the various use cases in which portions of the DTD are used.

2.2 Encoding of Enumerated Values

There are several places where either CDATA attributes or #PCDATA content must take on one of several enumerated types.

DECODES handles enumerations in a manner which may be useful for NWIS code.

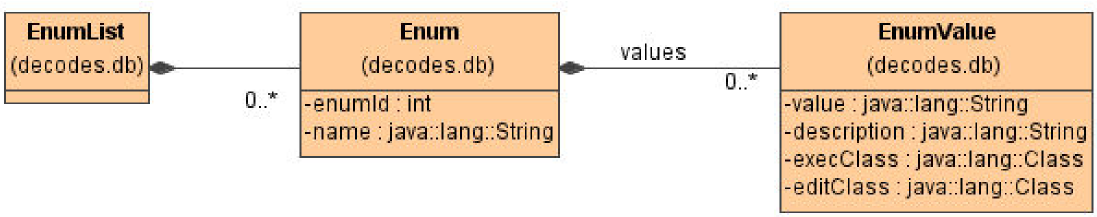

Figure 1: DECODES Enumeration Classes.

An “Enum” is a set of related choices. For example “DataTypeStandard” is an Enum. Its EnumValues are “SHEF”, “EPA”, and “NOS”.

Each EnumValue has the following:

- value: A unique string. This should be a name with no embedded blanks.
- description
- execClass (optional): A fully-qualified Java class name that instantiates objects of this type.
- editClass (optional): A Java class name for GUI editing values of this type.

Another example will illustrate how Enumerations are used. The following is the database file containing a list of output formatters for DECODES:

```
<Enum Name="OutputFormat">
  <EnumValue EnumValue="shef">
    <Description>Standard Hydrometerologic Exchange Format</Description>
    <ExecClass>decodes.consumer.ShefFormatter</ExecClass>
  </EnumValue>
  <EnumValue EnumValue="stdmsg">
    <Description>USGS Standard Message Format</Description>
```

```
      <ExecClass>decodes.consumer.StdmsgFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="emit-ascii">
      <Description>Compatible with EMIT ASCII format</Description>
      <ExecClass>decodes.consumer.EmitAsciiFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="shefit">
      <Description>USACE HEC Intermediate SHEF Format</Description>
      <ExecClass>decodes.consumer.ShefitFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="dump">
      <Description>Testing and trouble-shooting Format</Description>
      <ExecClass>decodes.consumer.DumpFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="ascii-table">
      <Description>Delimited row-column format</Description>
      <ExecClass>decodes.consumer.TableFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="emit-oracle">
      <Description>Compatible with EMIT Oracle format</Description>
      <ExecClass>decodes.consumer.EmitOracleFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="eumsg">
      <Description>USGS Engineering Unit Message Format</Description>
      <ExecClass>decodes.consumer.EumsgFormatter</ExecClass>
    </EnumValue>
    <EnumValue EnumValue="human-readable">
      <Description>Display Format</Description>
      <ExecClass>decodes.consumer.HumanReadableFormatter</ExecClass>
    </EnumValue>
  </Enum>
```

The “OutputFormat” Enum contains several values. Each one specifies an ExecClass. When the user wants to output data in the USGS EUMSG format, an object of type “decodes.consumer.EumsgFormatter” will by instantiated. There is no hard-coded list of classes for output formatters.

Thus it is very easy to extend DECODES functionality through enumerations. Similar lists are implemented other parameters, including::

- TableLookupAlgorithm - How to convert an input value through table-lookup, linear interpolation, exact-match, etc.
- DataConsumer - How data is stored after conversion.
- DataSource - How to retrieve raw data for conversion.
- DecodingScript - How to convert raw data to time-tagged engineering units.

In DECODES XML files, an enumeration value is simply CDATA. The checking for a valid value is done in the Java code by attempting to look up the value in the 'Enum' and 'EnumValue' tables. This makes it easy to implement new types as the code evolves.

In DECODES, selection of an Enum can cause the code to instantiate different classes (see the EnumValue.execClass member). This makes it easy to extend the code as new functionality is added. Perhaps NWIS could benefit from this scheme as well.

Hard-coding enumeration values within the DTD may cause the following problems:

- Upper/lower case must match exactly - Java code can be more flexible.
- Difficult to evolve code, especially if multiple copies of the DTD are out there.
- Hence you might end up with an XML file that is valid in one location, but won't parse in another.
- Error handling of an invalid enumeration value is better handled within the Java code. If done in the DTD, the whole document is invalid.

2.2.1 Reflist Class in NWIS

The NWIS Java software has already moved in a similar direction as DECODES. It implements Reference Lists using the class "Reflist". The "Reflist" class is then sub-classed for each place it is used.

For example, the class "TimeZoneList" in the business.dateTime package extends "Reflist".

The differences between NWIS Reflist and DECODES Enum are:

- NWIS Reflists are populated by Java code. DECODES Enums are stored in the database and are thus easier to extend.
- NWIS Reflist is an abstract base class that must be subclassed everywhere it is used. DECODES Enum is a complete class, not intended for extension.
- Consequently, each NWIS Reflist can store different kinds of things in its list and can exhibit different behavior.

Recommend:

1. Store NWIS Reflists in the NWIS SQL database and also establish an XML format. **We agree, and this has been done**.

2. Try to come up with a common subset of functions that all Reflists must implement. Put this in the Reflist base class. Only use subclassing where necessary. **We agree.**

3. Extend DECODE EnumValue with the addition of sort order. Also add a DefaultValue attribute to the Enum entity (and SQL Enum table). **We agree.**

2.3 Date Values

Many elements contain date values. These are defined as CDATA attributes in the DTD. The NWIS code appears to support a multitude of complete and partial date/time formats:

```
public static String[] DATEFORMATS = {
            "mmddyyyy", "mmyyyy", "yyyy",
            "mm dd yyyy",  "m dd yyyy", "mm d yyyy", "m d yyyy",
            "mm/dd/yyyy",  "m/dd/yyyy", "mm/d/yyyy", "m/d/yyyy",
            "mm-dd-yyyy", "m-dd-yyyy", "mm-d-yyyy", "m-d-yyyy",
            "MMM dd yyyy", "MMM d yyyy",
            "yyyymmdd", "yyyymdd", "yyyymmd", "yyyymd",
            "yyyy mm dd", "yyyy m dd", "yyyy mm d", "yyyy m d",
            "yyyy.mm.dd", "yyyy.m.dd", "yyyy.mm.d", "yyyy.m.d",
            "yyyy_mm_dd", "yyyy_m_dd", "yyyy_mm_d", "yyyy_m_d",
            "dd-MMM-yyyy", "d-MMM-yyyy"};
public static String[] PARTIALDATEFORMATS = {
                     "yyyy", "mmyyyy", "myyyy",
                     "yyyy", "mm yyyy", "m yyyy",
                     "mm/yyyy", "m/yyyy",
                     "mm-yyyy", "m-yyyy",
                     "MMM yyyy",
                     "yyyymm", "yyyym",
                     "yyyy mm", "yyyy m",
                     "yyyy.mm", "yyyy.m",
                     "yyyy_mm", "yyyy_m",
                   "MMM-yyyy"};
public static String[] TIMEFORMATS =
  {"hhmmss","hh:mm:ss", "hhmm", "hh:mm"};
```

2.4 DTD Validation

The HydroML DTD currently resides at:

```
http://water.usgs.gov/nwis_activities/xml/Hydroml.dtd
```

Hence, valid Hydroml files should include the line:

```
<!DOCTYPE Hydroml SYSTEM "http://water.usgs.gov/nwis_activities/xml/Hydroml.dtd">
```

This states that the root element is always expected to by “Hydroml”, which is defined by the DTD at the stated URL.

Recommend: Implement the file parser in such a way that validation can be turned off if necessary. For example, we might want to do this for stand-alone processing on a system that is not connected to the internet. This can be accomplished by setting the feature: “http://xml.org/sax/features/validation” to a value of false. We agree.

Note: Features supported by Apache Xerces are defined at:

```
http://xml.apache.org/xerces-j/features.html
```

2.4.1 Strategies for Validation

Syntax vs. Content

Syntax checking should clearly be done in a DTD. This would include:

- Listing the allowable children and attributes of each element
- Defining the cardinality of child elements.

Attribute format checking (dates, lat/long, numbers, etc.) should be done in Java code. This will allow more control in error handling & recovery.

Enumerations

Enumerations are often spelled out in the current DTD.

Recommend: Consider leaving some enumerations defined as CDATA, to be validated by Java code. This will allow for easier future expansion. For example, encoding is defined as an enumeration of either “Text” or “base64”. Leaving this as CDATA could allow for easier implementation of new encoding types in the future. We agree with the exception of the encoding, we feel that should remain within the XML definition and new additions can be added easily.

In other words, only use DTD-defined enumerations in places where you are absolutely sure there will never be a need for expansion.

The “ID” and “IDREF” types

Some entities uses a DTD “ID” data type. This has the following drawbacks:

- IDs are constrained as to content. They must be valid "XML Names" and cannot start with an initial digit. I assume you want to use this for a USGS Site Number.
- No other ID attribute in the XML document can have the same value. For example, what about a file that contains multiple records from the same site, perhaps a kind of site-history?

Recommend: Stay away from the ID and IDREF types in DTD. Do referential integrity checking in Java code. Use NMTOKEN where a single unique name is to be provided. **We agree and this has been changed.**

3 HydromI Database Elements

Starting at the top of the hierarchy, as defined in the DTD, this section lists the elements other than SiteVisit. Most of these elements will live in the SQL database and will be more or less persistent.

3.1 Top Level

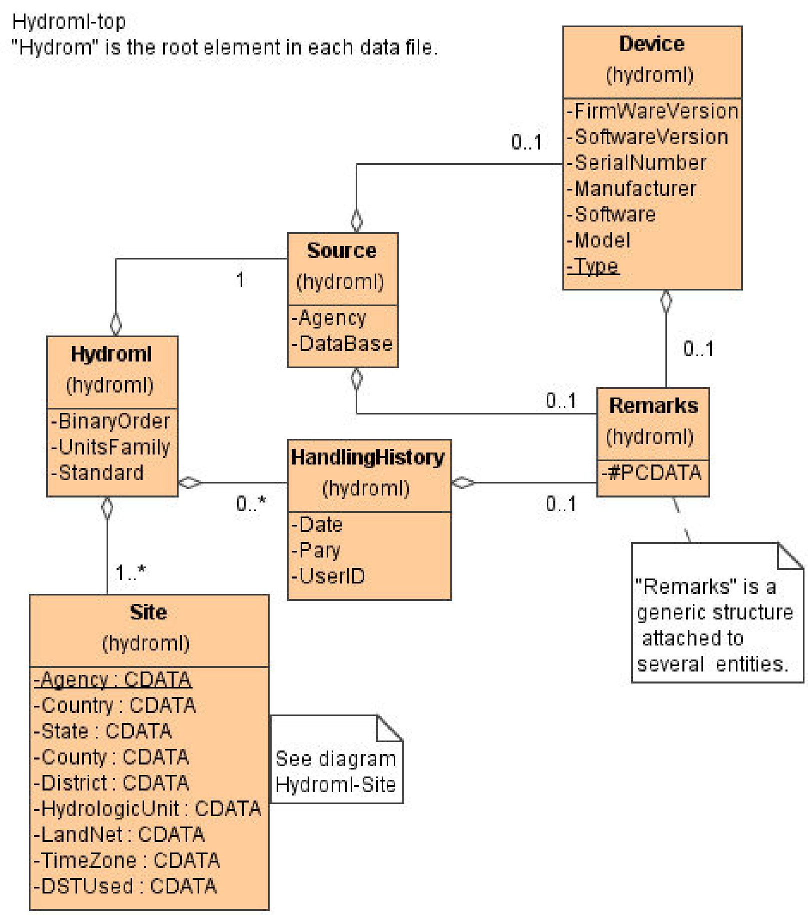

Figure 2: HydromI-TopSite Information

Notes on HydromI content relating to SiteVisit files produced in the field:

- Source elements in a SiteVisit file will most likely contain only the Agency name. Even this may be omitted and a default value of "USGS" will be supplied.
- Source elements in a SiteVisit file will not contain subordinate Device records.
- The HandlingHistory and its subordinate Remarks will not normally be included in files coming from the field. These are to be filled in by the Ingest process. They will be included in the Archive XML files produced as part of the ingest process.
- If a HandlingHistory record *is* included in a field-generated SiteVisit file, it will be overwritten.

3.2 Sites

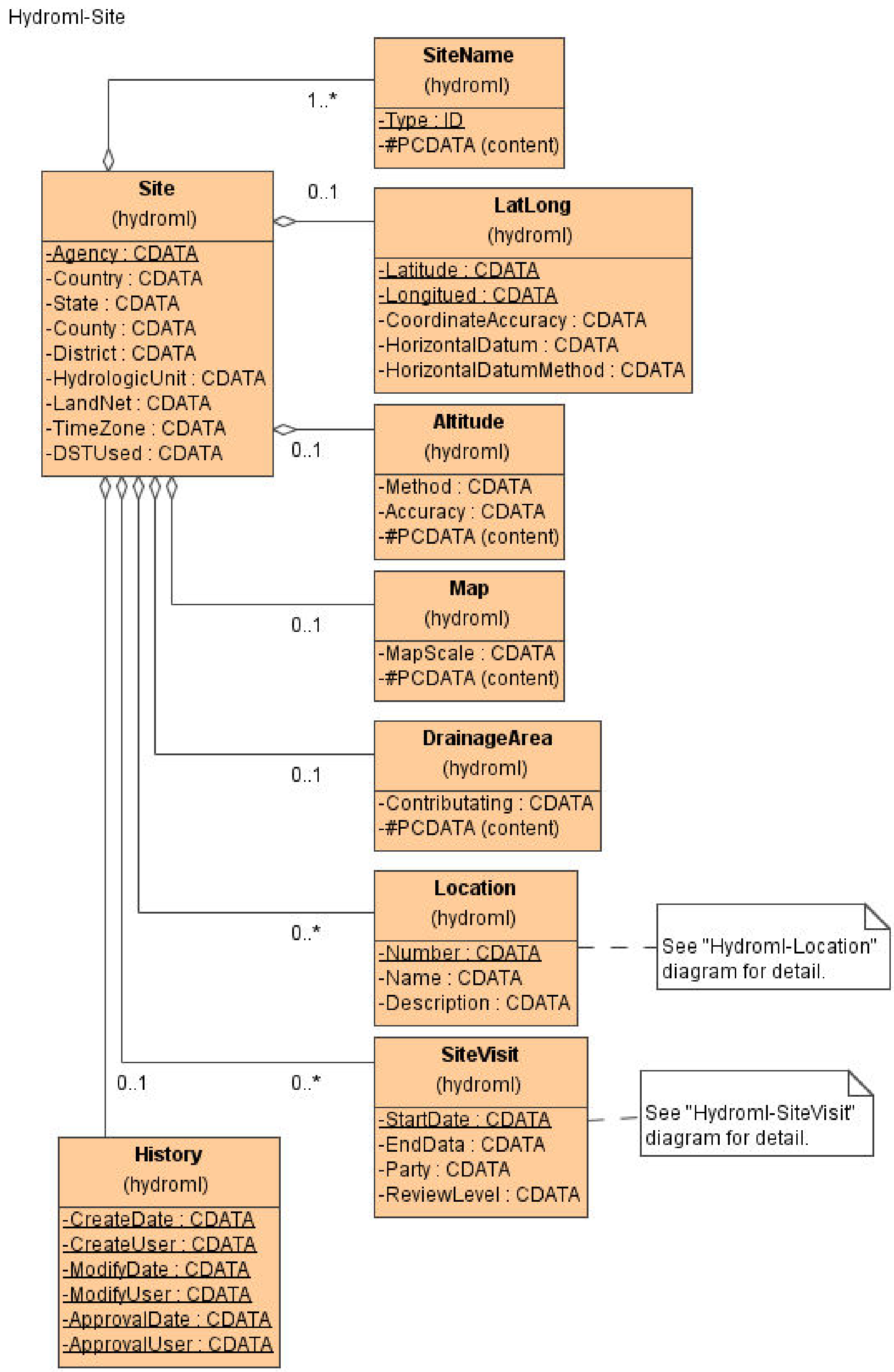

Figure 3: Hydroml-Site

Recommend: Use NMTOKEN rather than ID for SiteName "Type" attribute. **We agree and this has been changed.**

3.2.1 Site Information Overlap with DECODES

One enumeration value for Site Name Type is specified as NWSWB4 for NWS Work Book 4 ID. Workbook 4 has been superceded by AFOS Handbook 5.

Recommend: Change this Site Name Type to "NWSHB5". This will make it compatible with DECODES. **We agree and this has been changed.**

In DECODES a "Site" is the finest granularity for specifying a location. In NWIS, a Site can have zero or more Locations associated with it (e.g. for different gaged channels, or different gages.) In NWIS a Site may have a default Location, which is usually all that is necessary. This is an unfortunate difference in semantics that we may simply have to live with.

3.3 Location Data

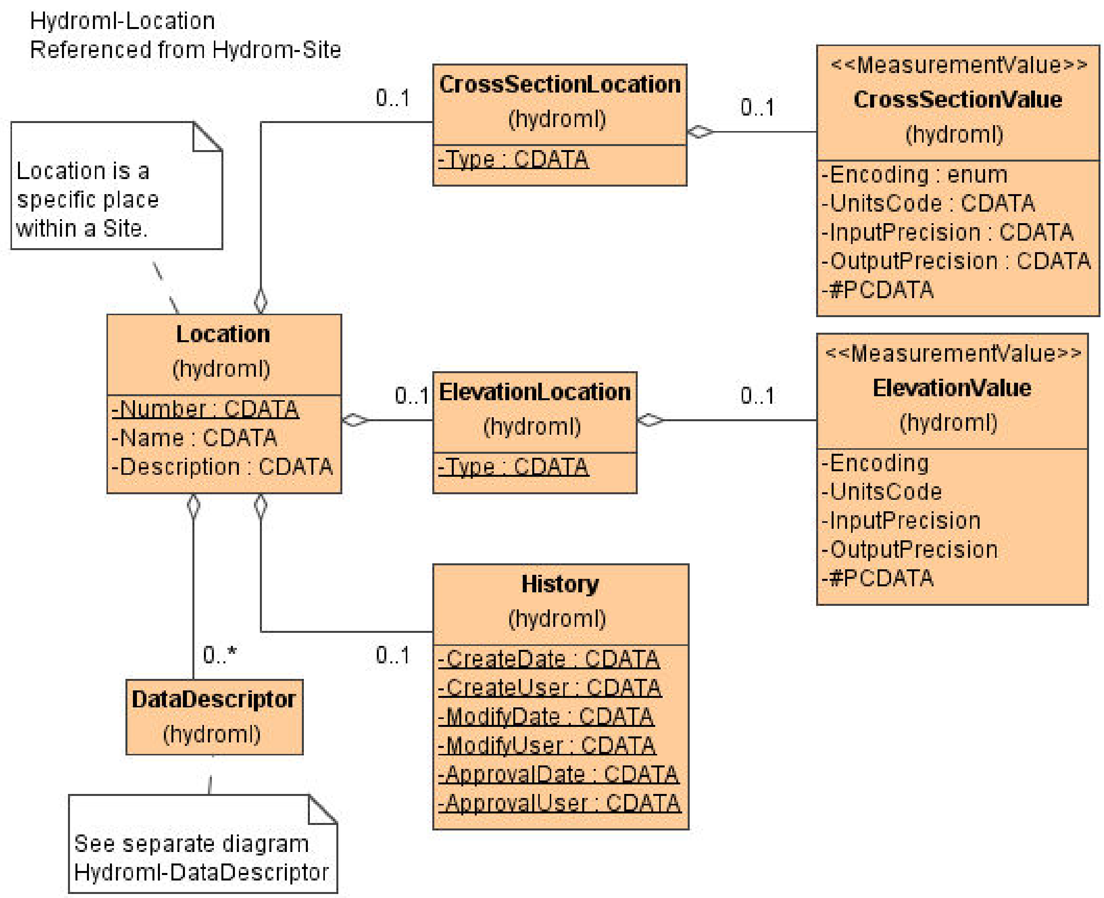

Figure 4: Hydroml-Location

CrossSectionLocation and ElevationLocation each take a “Type” attribute. This describes the context and reference point for the location. For example, “On Left Bank” would not require a CrossSectionValue. “From Left Bank” would require an associated value.

ElevationLocation type would specify the reference point for the elevation value.

3.4 Data Descriptor Information

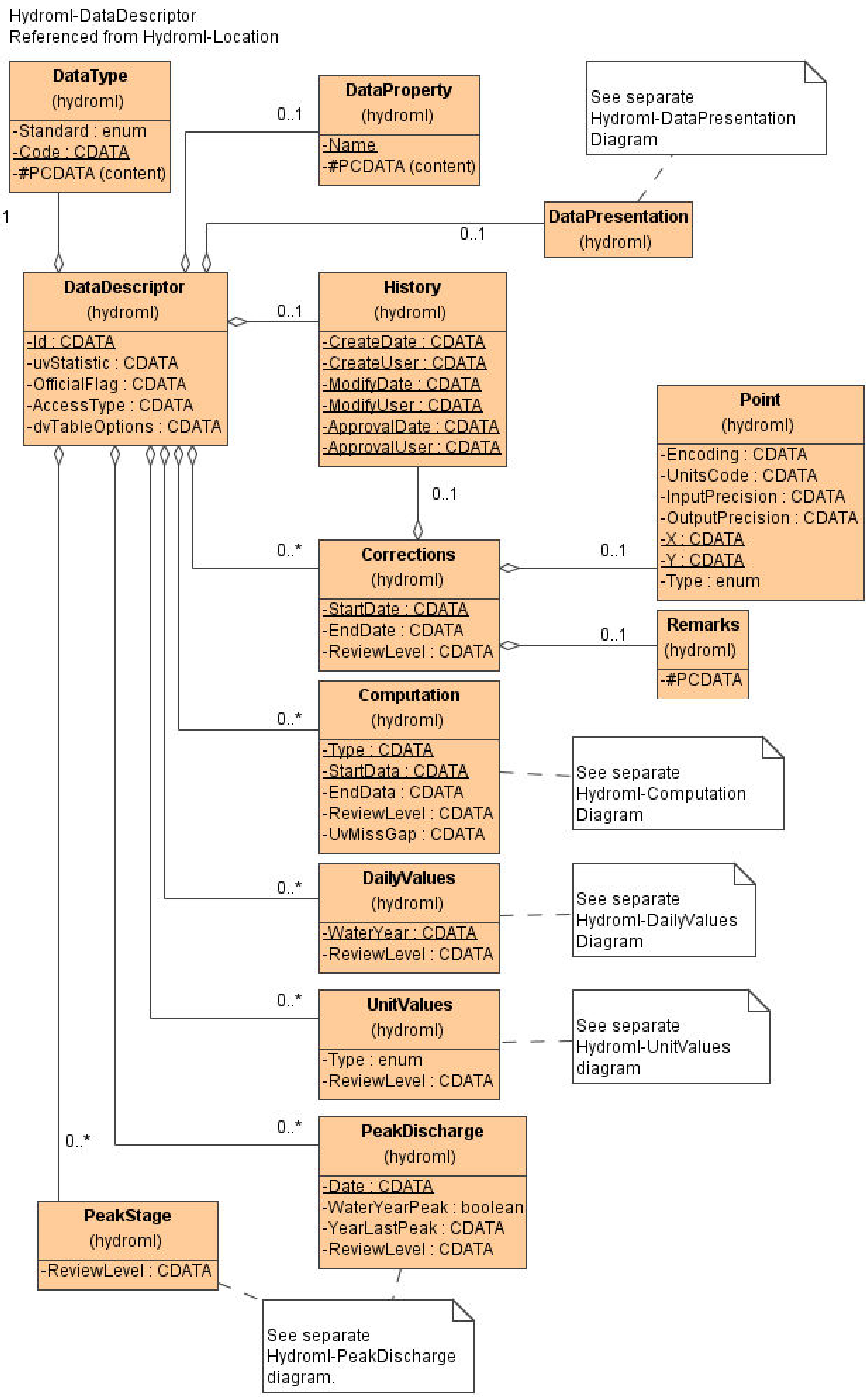

Figure 5: Hydroml-DataDescriptor.

Recommend: Remove the #PCDATA content from DataType. As in DECODES, a DataType should have EMPTY content. **We agree and this has been changed.**

The attribute list for DataProperty is:

```
<!ATTLIST DataProperty Name  (DDID) #REQUIRED >
```

This means that the Name can only take on the literal value "DDID". Surely this is not the intent.

Recommend: Declare DataProperty Name attribute as NMTOKEN. **We agree and this has been changed.**

Notes:

1. The DataDescriptor element is identified by the DDNO used within DECODES.

2. None of the information at or below DataDescriptor will be included in a SiteVisit Hydroml file.

3. DataDescriptor will be included in Hydroml files distributed to other agencies who need the USGS Computations, such a stage-to-flow conversions.

3.5 Data Presentation Information

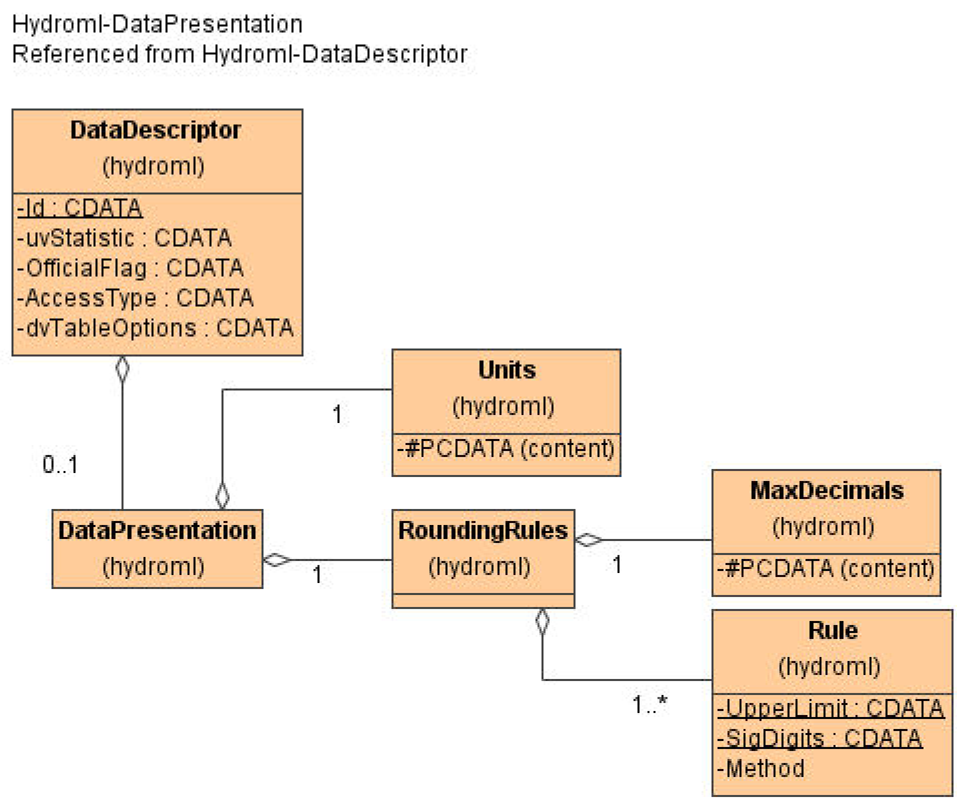

Figure 6: Hydroml-DataPresentation

Recommend: Get rid of Units entity. Rather make "UnitsCode" a CDATA attribute of DataPresentation. This makes it consistent with many other entities. We agree and this has been changed.

The RoundingRules hierarchy appears to be modeled on the overly-flexible DECODES model. I prefer the 10-character string used in the NWIS java code.

Notes about rounding rules from discussions with USGS:

- A RoundingRules entry contains a set of Rules and an absolute MaxDecimals indicator. The MaxDecimals may override the Rule
- Each Rule contains an upper limit, number of significant digits and a 'method' indicator.
- This Rule would apply to values equal to, or below the upper limit, but greater than the upper limit of the next lower rule.
- Method is 'Standard' or 'Midpoint'. This should be specified in the DTD as CDATA. Other rounding methods may be implemented in the future.
- Midpoint rounding means that the least significant digit is rounded to either 0 or 5.
- The old 10-character string can be represented as a set of 10 rounding rules. The first element would have an upper limit of 0.01, the second woud be 0.1, etc.
- It is USGS's intent to migrate the NWIS code in this new direction.
- The DECODES rounding rules are not implemented quite right. They will need to be modified to conform to this.

3.5.1 Contrast with Data Presentation used by DECODES

Note the following hierarchy. See that NWIS takes a very granular and rigid approach to data presentation. Is this really necessary?

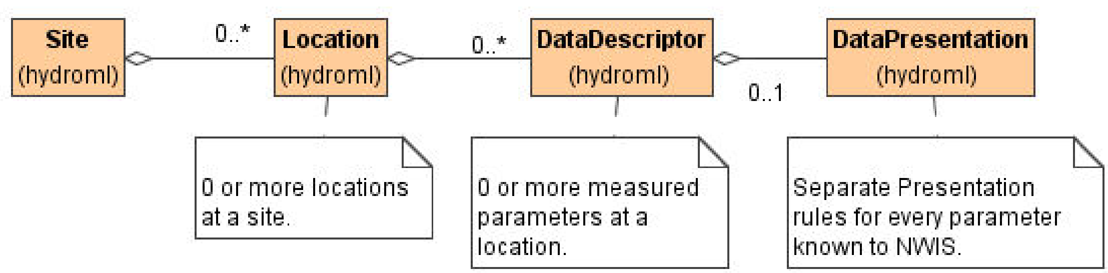

Figure 7: Data Presentation Hierarchy in NWIS.

In DECODES, data presentation information was purposefully separated from other data descriptors. This allows different groups of users to look at the same data values, but having different Data Presentations applied. For example, data from a watershed that spans the border between U.S. and Canada. Americans may want to view data in English units. Canadians may prefer metric units.

In DECODES "DataPresentation" elements are aggregated into "PresentationGroup" elements. They are not tied directly to every data descriptor and set of data values. Rather they may be applied after the fact.

In DECODES, a DataPresentation is associated with a DataType. For example, there is a PresentationGroup called "SHEF-English" that defines the presentation for all SHEF codes. For example, the group contains a DataPresentation element that states that stage values with SHEF code "HG" must be in feet according to the SHEF standard.

3.6 Measurements Values

Many elements in the hierarchy have the same structure as a generic "Measurement Value". This structure contains 4 elements:

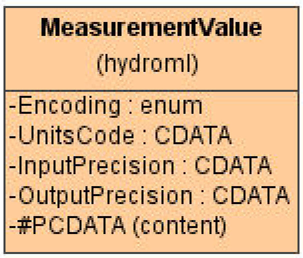

Figure 8: Measurement Value Structure.

Notes on Attributes:

- All attributes are optional and have defaults.
- Encoding default is "Text". May also be "base64".
- UnitsCode: Many of the units will be assumed based on the context. They may also be explicitely specified here. A possible reason for this would be to override the default "English" specified at the top-level Hydroml entity.
- InputPrecision: This is a single digit, intended to signify the significant figures if >= 1; or the number of decimal points if less than 1.
- OutputPrecision: Likewise, this is a single digit.

The #PCDATA content contains the actual value and is required.

Recommend: Don't hard code possible settings for "Encoding" in the DTD. Other encodings may need to be implemented in the future, including GOES DCS Pseudo-binary. We disagree, this will remain because it will always automatically include the default type and also it is easy to add new types of encodings.

USGS Requires that input and output precision be carried along with every measurement point. This will not be modified.

The InputPrecision may be applied to the value before it is used. However, we want to be able to capture the #PCDATA value that the user actually entered (may be more or less precision) so that it can be faithfully echoed.

Recommend: Use a Parameter Entity to define the content of a Measurement Value in the DTD. Use Parameter Entity References in each of the classes. This will simplify and shorten the DTD. It also allows you flexibility if the MeasurementValue structure ever needs to change. We agree and this has been changed.

Using the Parameter Entity will also simplify the DTD declaration of elements that supplement the basic "MeasurementValue" structure with additional attributes

The following Measurement Values add a "WaterYear" attribute:

- MonthlySummaryMaximum
- MonthlySummaryMinimum

The following Measurement Values add a "Date" attribute:

- MaximumDailyFlow
- MinimumDailyFlow
- Minimum7DayFlow
- MaximumInstantaneousFlow
- MinimumInstantaneousFlow
- MaximumInstantaneousStage

3.7 Computations

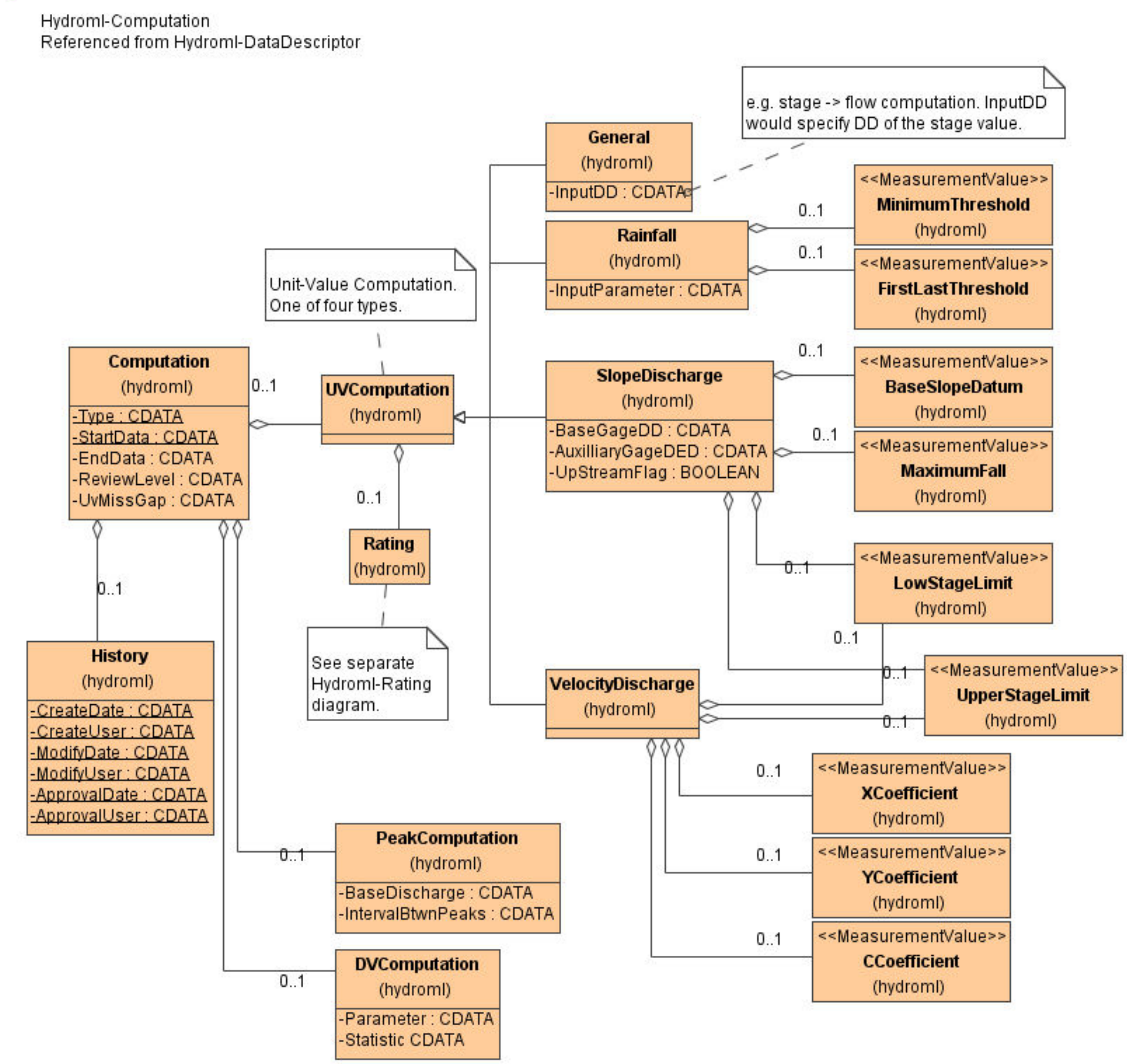

Figure 9: Hydroml-Computation.

Note: Computations are not part of a SiteVisit file. They would be part of a Hydroml used to exchange rating tables with other agencies.

An example for a stage to discharge computation:

- General.InputDD would be the DD for the stage value.
- The DataDescriptor parent would be the DD of discharge.
- The Input value is applied to the rating to produce an output.

Note: Rainfall is a special type of computation that does not require a rating.

3.8 Ratings

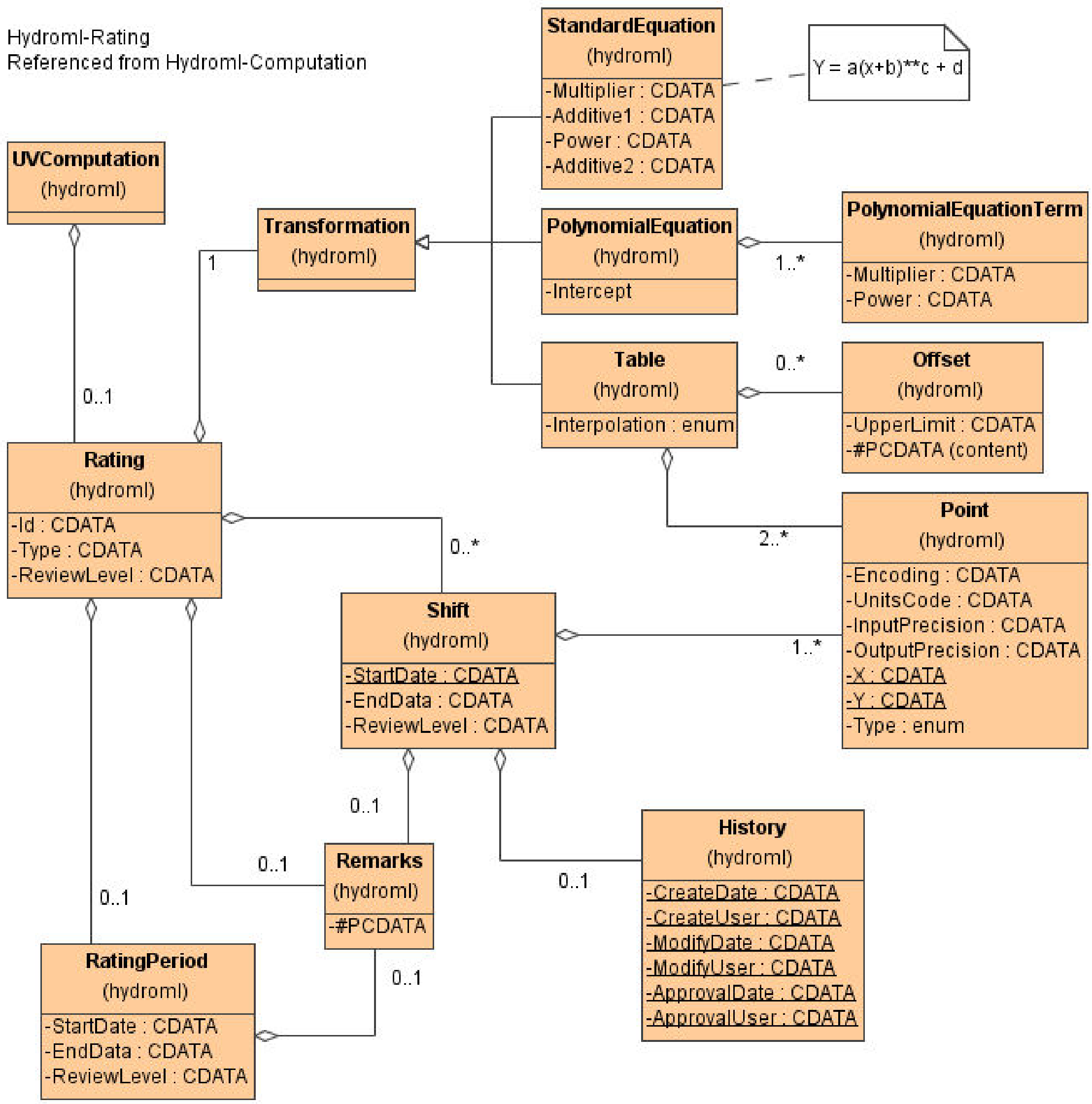

Figure 10: Hydroml-Rating.

Notes about Ratings:

- Ratings are a sub-element of Computations and as such will not be included in SiteVisit files.
- There must be at least 2 Points in a Table. It is not possible to state this in the DTD so it must be checked by the ingestor process.
- Ratings come and go as the stream contour changes. Hence the need for a rating period. Ratings remain in the NWIS database and may actually be re-activated at a future point.

Consider: A rating cannot be "re-activated" given the cardinality shown in the diagram because a rating can have only one RatingPeriod. Reactivation would involve making a copy of an old rating. This may be the simplest way to implement re-activation, especially if it is a rare occurrence. ***This is wrong because a rating can be re-activated and can have multiple start and end dates. The XML DTD has been modified to allow multiple rating dates.***

Recommend: Within PolynomialEquationTerm, Multiplier and Power should be declared as #REQUIRED? **We agree and this has been changed.**

The NWIS XML represents standard and polynomial equations along with Table lookup operations. These are implemented as specializations of a general "Transformation" class.

3.8.1 DECODES Engineering Unit and Raw Conversions

The following diagram shows the structure of Engineering Units (EU) and Conversions in the DECODES database:

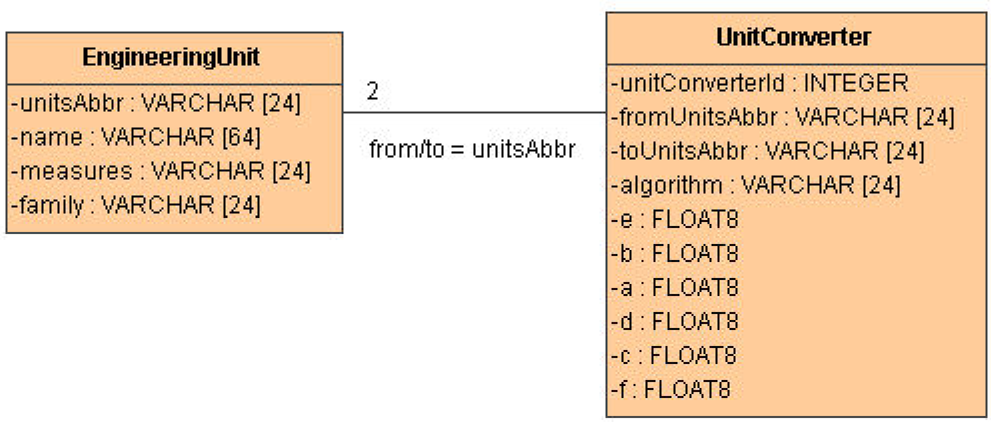

Figure 11: DECODES Engineering Units & Conversions Data Structures.

Each EU record contains:

- UnitsAbbr - Unique abbreviation used to reference this EU.
- Name - Complete descriptive name
- Family - English or Metric
- Measures - What physical property is being measured (e.g. length, volume).

The following shows two XML records defining the units "mG/L" and "cc":

```
<EngineeringUnit UnitsAbbr="mG/L">
  <Name>milligrams per liter</Name>
  <Family>Metric</Family>
  <Measures>concentration</Measures>
</EngineeringUnit>
<EngineeringUnit UnitsAbbr="cc">
  <Name>cubic centimeter</Name>
  <Family>Metric</Family>
  <Measures>volume</Measures>
</EngineeringUnit>
```

Each UnitConverter record contains:

- From and to EU abbreviations.
- An algorithm (see below)
- Up to 6 coefficients

DECODES uses UnitConverter records for two purposes:

- As part of a decoding script, a converter takes raw values to convert to a known engineering unit (EU) value. In this case the “from” abbreviation is always “raw”.
- In post-processing, to convert from one EU value to another (e.g. inches to cm). This is the database of known conversions.

DECOES database and XML files use a compact representation for UnitConverter equations. DECODES currently supports 4 algorithms:

- “None” — Not really an algorithm. Used to express that no conversion is necessary. For EU conversions, this implies that two different units are synonyms (e.g. “cc” and “ml”).
- “Linear” — y = Ax + B. User supplies A & B coefficients. This handles most EU and raw conversions.
- “USGS-Standard” — y = A (x + B) ^C + D.
- “Poly5” — y = Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F

For example, this record is part of a decoding script that converts raw values to Volts:

```
<UnitConverter FromUnitsAbbr="raw" ToUnitsAbbr="VOLTS">
  <Algorithm>linear</Algorithm>
  <A>0.0588</A>
  <B>0.0</B>
</UnitConverter>
```

The software reads the records and instantiates the appropriate type of conversion. A delegation pattern is used. The following diagrams the Java code:

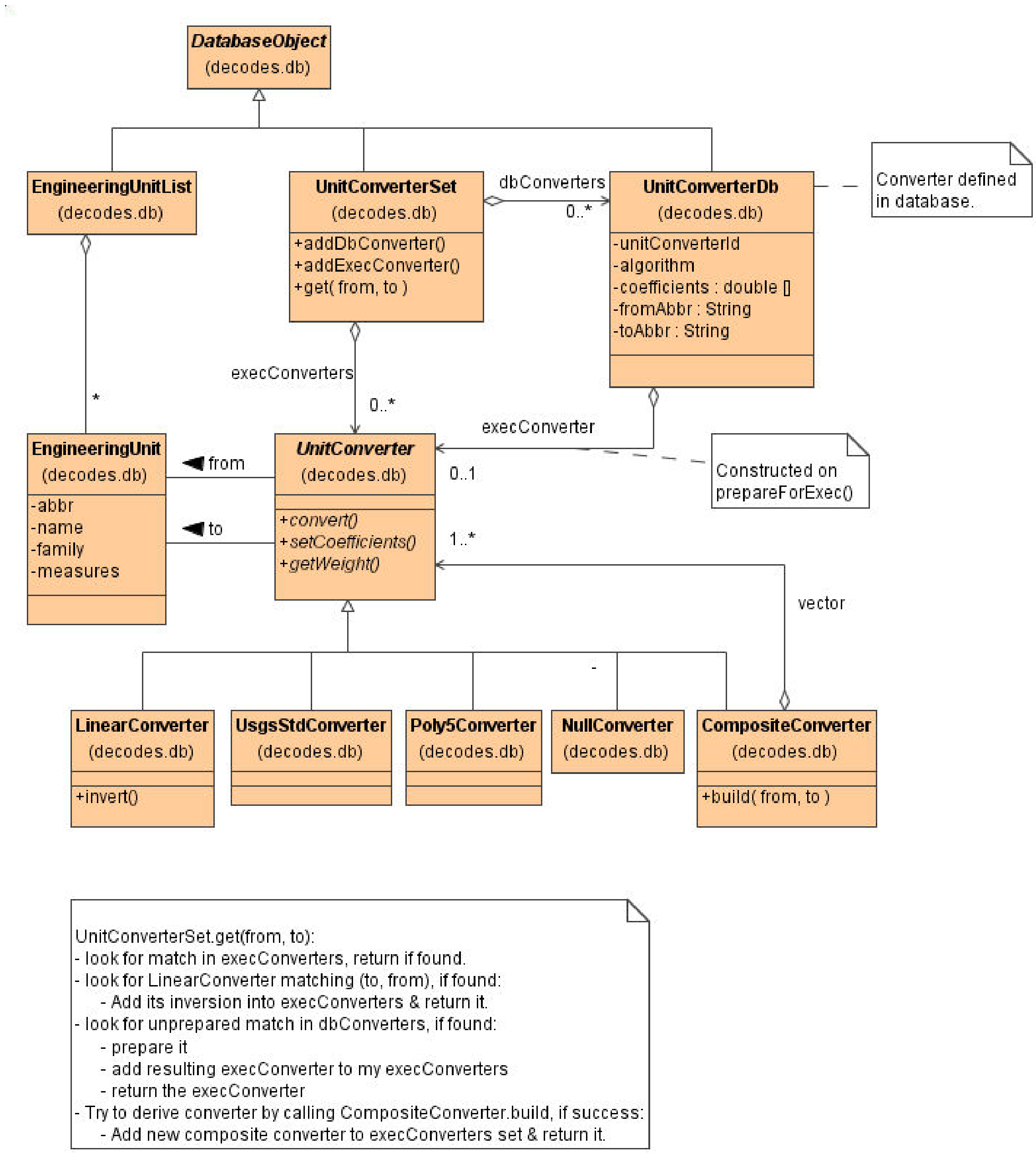

Figure 12: DECODES Java Code for Engineering Unit Conversions.

The software can automatically combine multiple conversions. For example, if it needed to convert inches to centimeters, the software might combine the following database-defined conversions:

- inches to feet
- feet to meters
- meters to centimeters

The software will use database-defined records to find the most efficient composite conversion.

3.9 Daily Value and Statistics Information

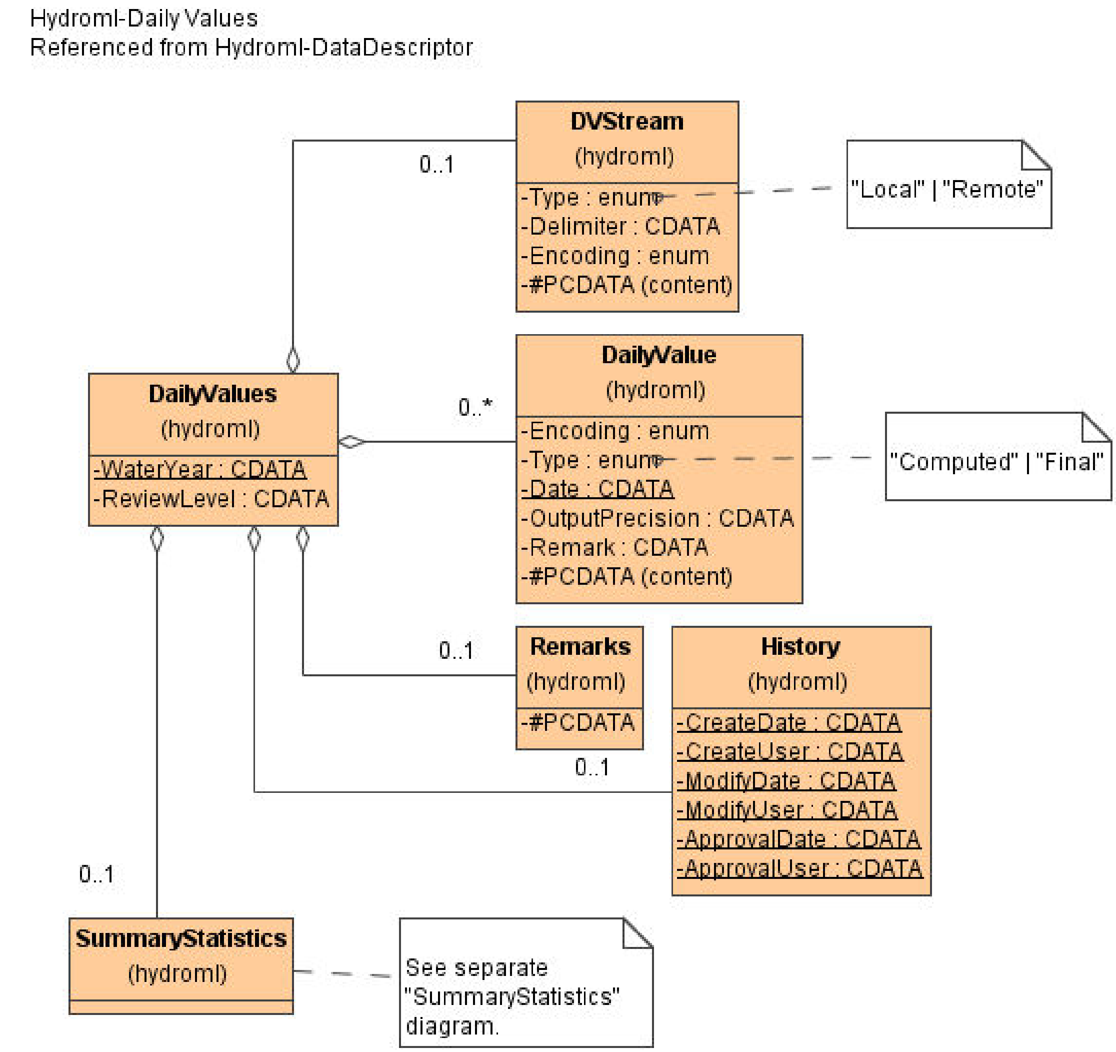

Figure 13: Hydroml-DailyValues

DailyValues and subordinate elements will not be part of a SiteVisit file. These would only be included in Hydroml files created as an NWIS export.

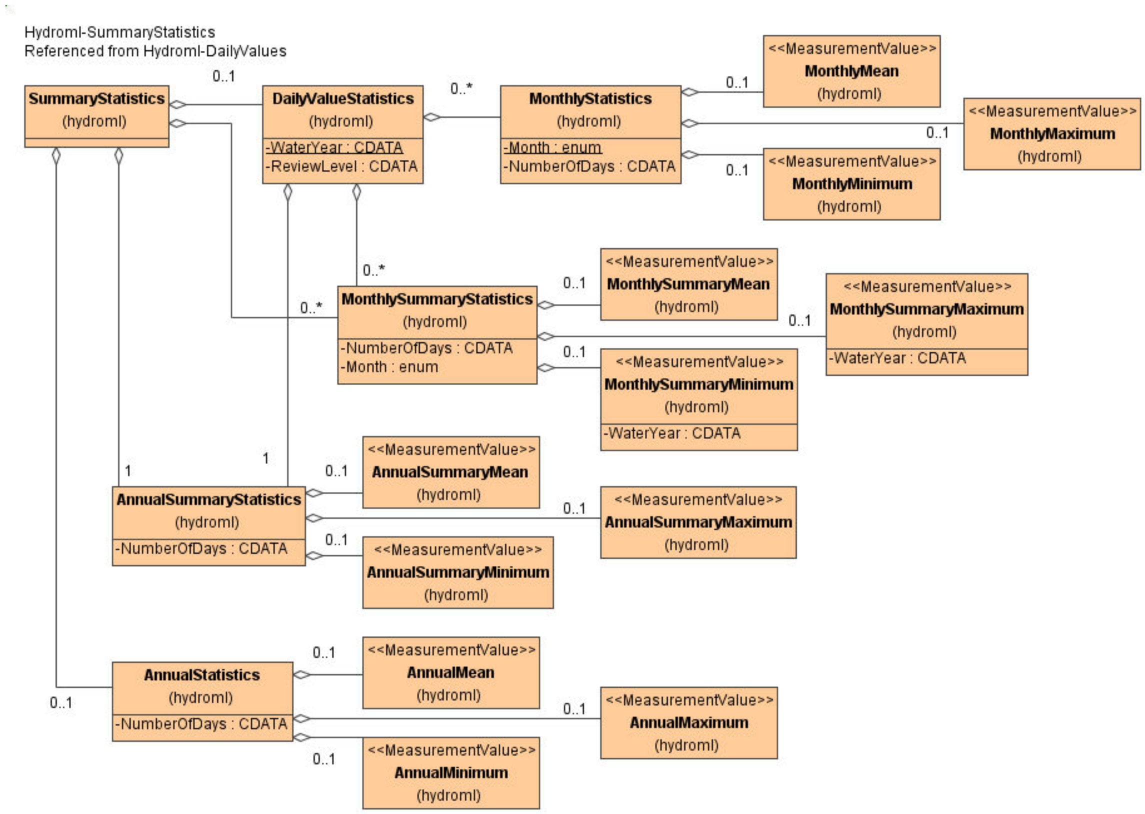

Figure 14: Hydroml-SummaryStatistics.

SummaryStatistics and subordinate elements will not be part of a SiteVisit file. These would only be included in Hydroml files created as an NWIS export.

For USGS to consider: What is the rational for the multiple paths to the AnnualSummaryStatistics and MonthlySummaryStatistics structures? ***This was a mistake in the DTD and has been corrected by removing the element SummaryStatistics which makes the DailyValuesStatistics element the top level element for this data.***

Note: AnnualStatistics are for a single year. AnnualSummaryStatistics will cover multiple years.

3.10 Discharge Period Statistics

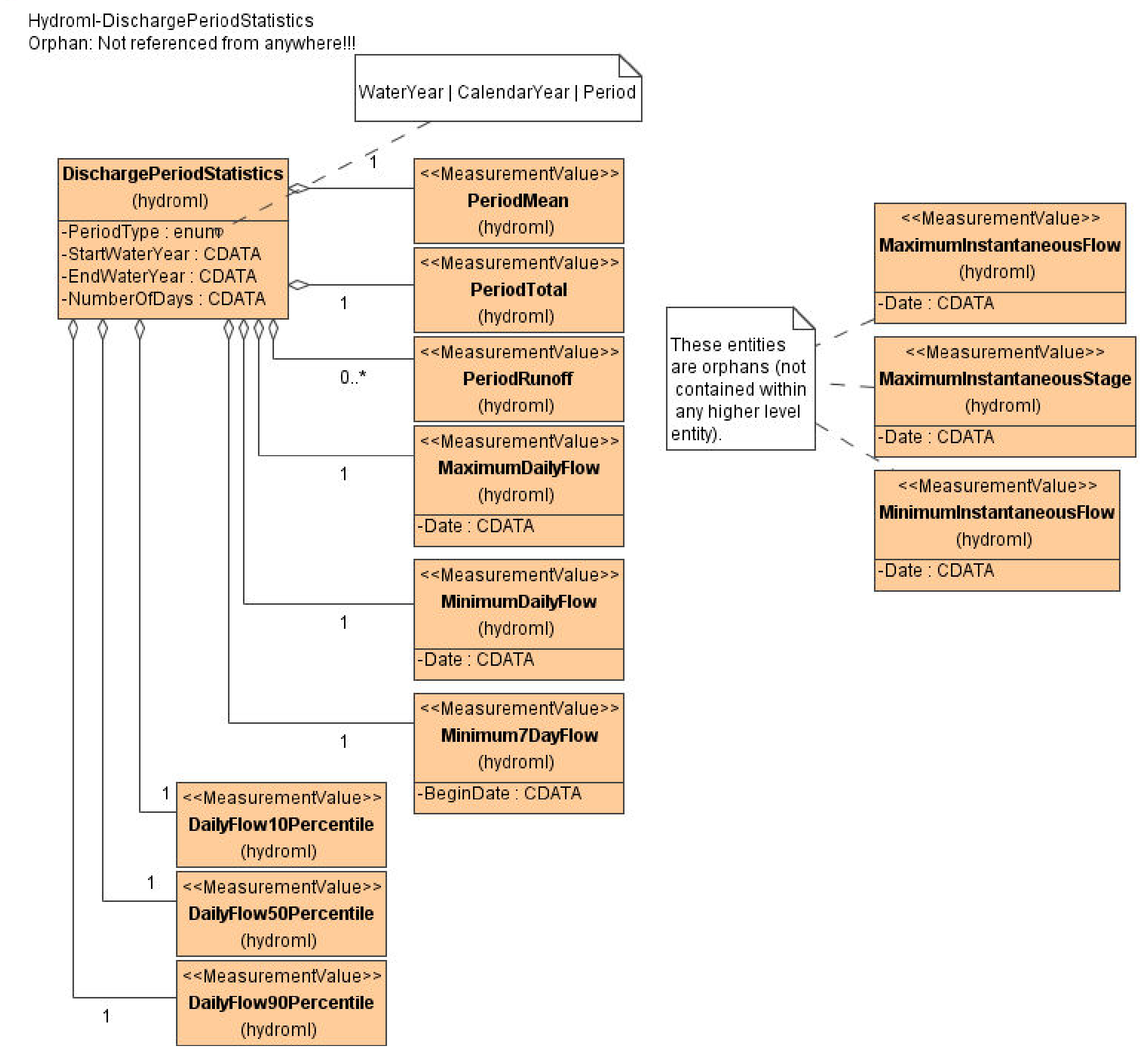

Figure 15:Hydroml-DischargePeriodStatistics.

DischargePeriodStatistics and subordinate elements will not be part of a SiteVisit file. These would only be included in Hydroml files created as an NWIS export.

Note, the Date attribute included in Max and Min elements are the estimated date of the maximum or minimum within the period covered by the statistics.

*Consider: DischargePeriodStatistics is an orphan. It is not referenced in any higher-level entities. What is the rationale for this**? This has been corrected by including it in the element DailyValuesStatistics.***

Minimum7DayFlow's attribute is "BeginDate". This is perhaps more descriptive but it makes it harder to implement a common parser.

3.11 Unit Values

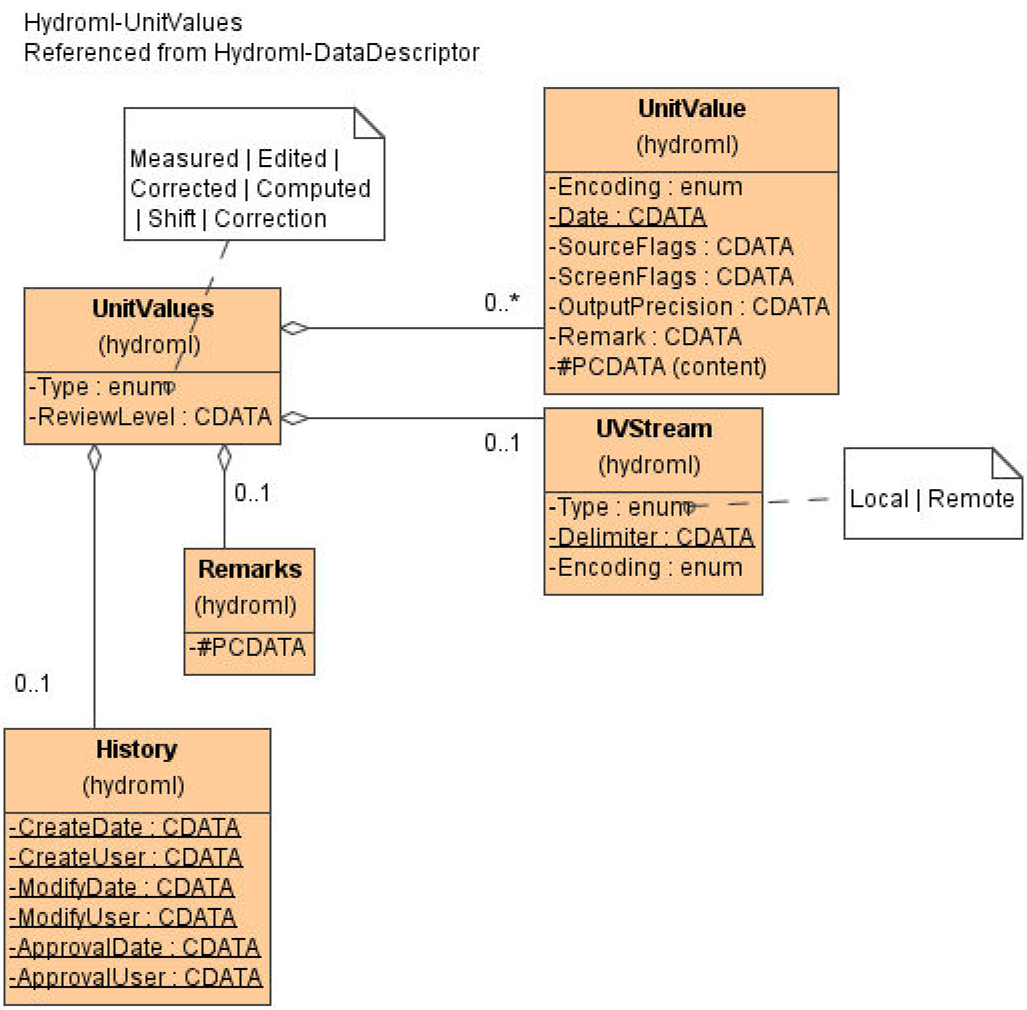

Figure 16: Hydroml-UnitValues.

UnitValues and subordinate elements will not be part of a SiteVisit file. These would only be included in Hydroml files created as an NWIS export.

Note: The UnitValues are populated from values received from DECODES.

UVStream is just a string encoding of an array of UnitValue entities.

Consider: Why are there two separate ways of representing an array of unit values? What are the use-cases for each type of structure?

Recommend: Allow only one type of representation unless there is a compelling reason to do otherwise. We disagree because we feel this gives us additional flexibility to output unit values data.

3.12 *Peak Discharge and Peak Stage Information*

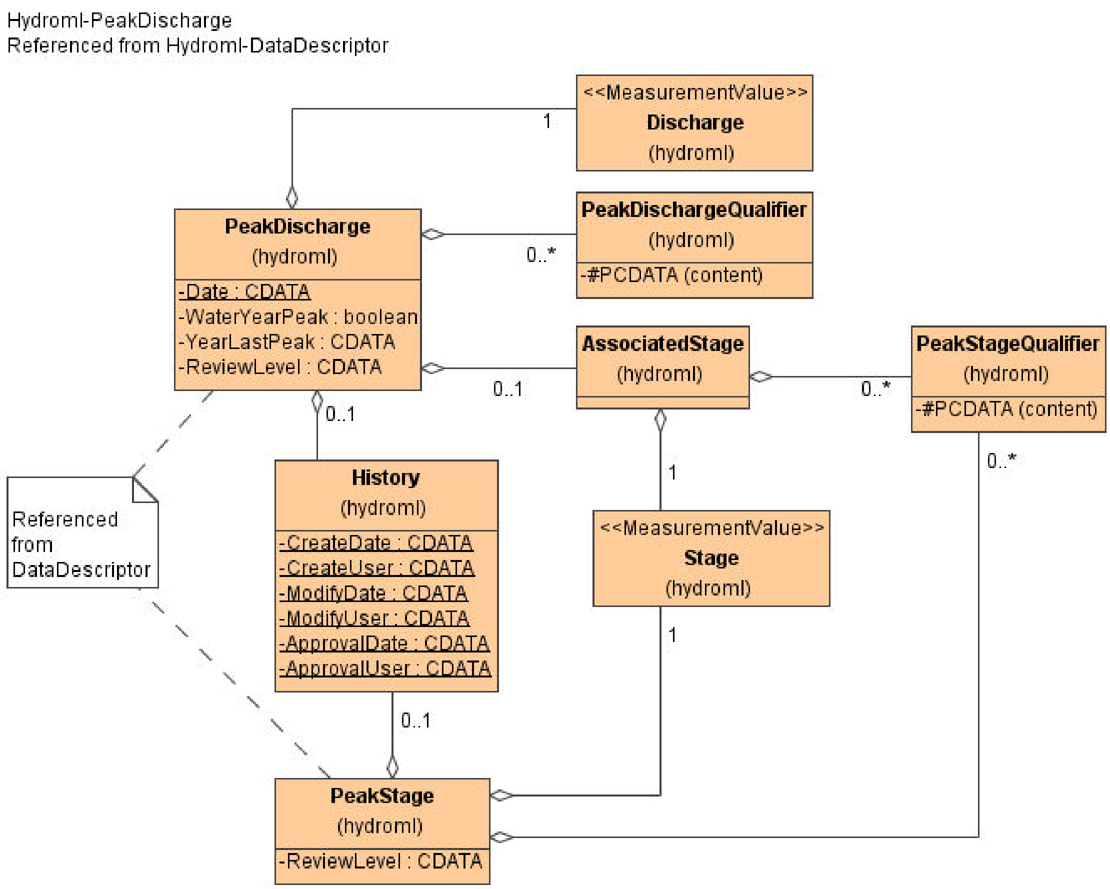

Figure 17: Hydroml-PeakDischarge.

Peak Discharge and subordinate elements will not be part of a SiteVisit file. These would only be included in Hydroml files created as an NWIS export.

4 Site Visit Information

Site Visit information is of prime importance to the use-cases for this project. Site Visit represents about the bottom 35% of the Hydroml hierarchy.

4.1 SiteVisit Top-Level

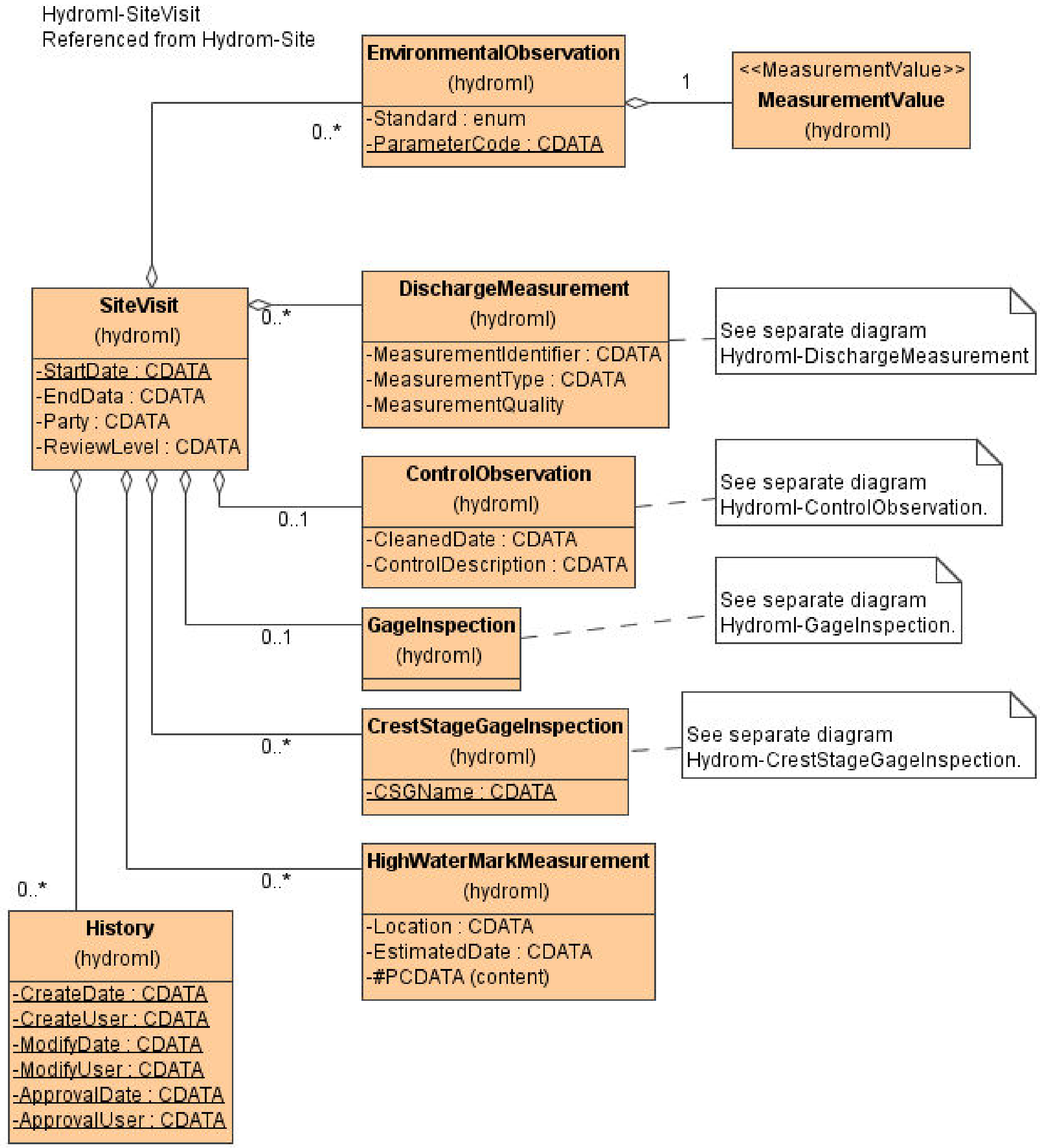

Figure 18: Hydroml-SiteVisit.

Recommend: EnvironmentalObserveration should contain DataType, rather than have these attributes. This will make them consistent with other unit values in the Hydroml. We agree and have made the modification.

Environmental observation elements are for miscellaneous observations made at the site for air temperature, relative humidity, etc. Each would be represented by a data type element and a measurement value.

4.1.1 Discharge Measurement

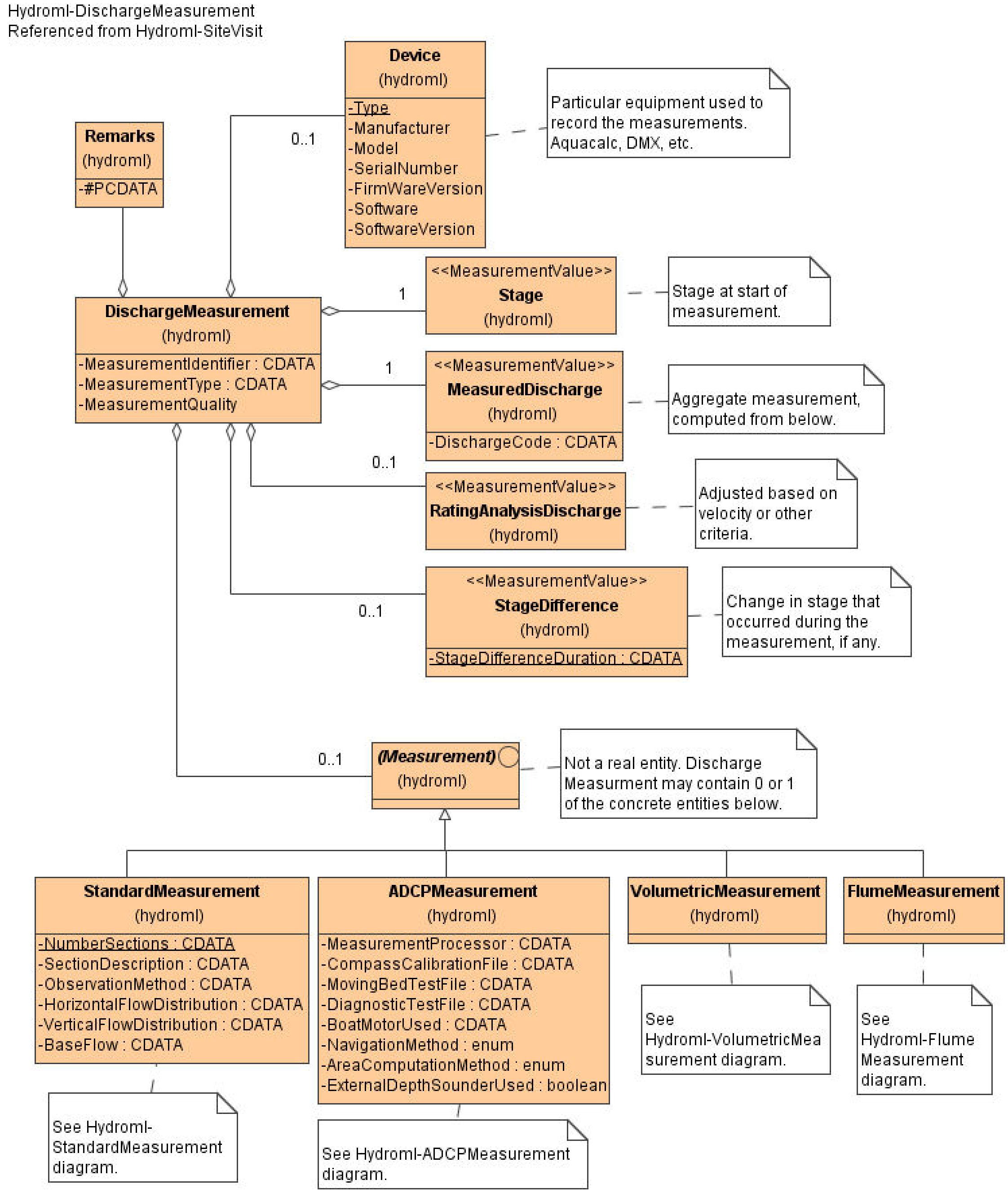

Figure 19: HydromI-DischargeMeasurement.

The "DischargeMeasurement can contain one of the 4 measurement types. This is shown in the diagram via a (nonexistent) "Measurement" interface.

4.1.2 Standard Measurement

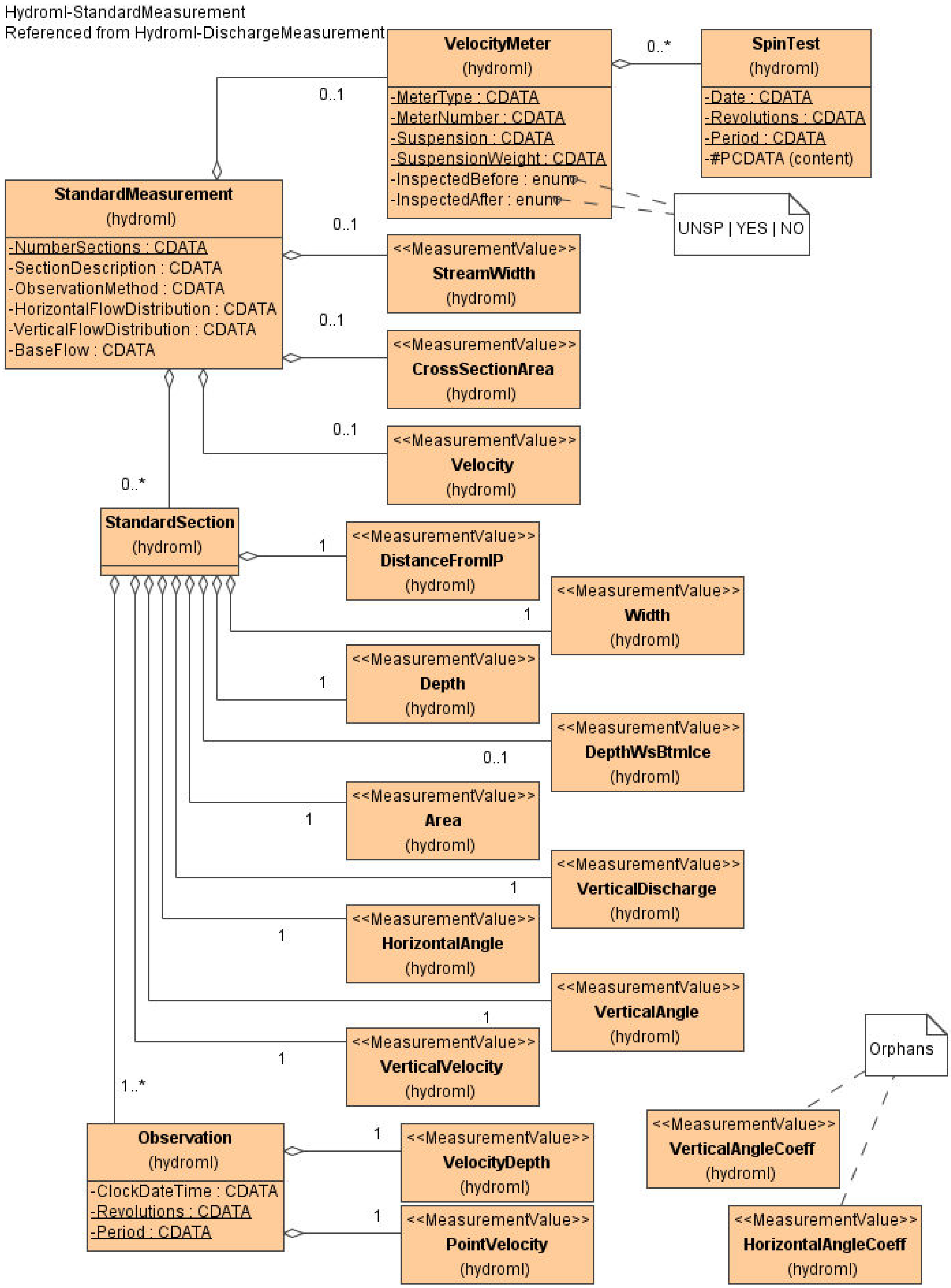

Figure 20: HydromI-StandardMeasurement.

For context, the following diagram shows where StandardMeasurement occurs within the entire file hierarchy:

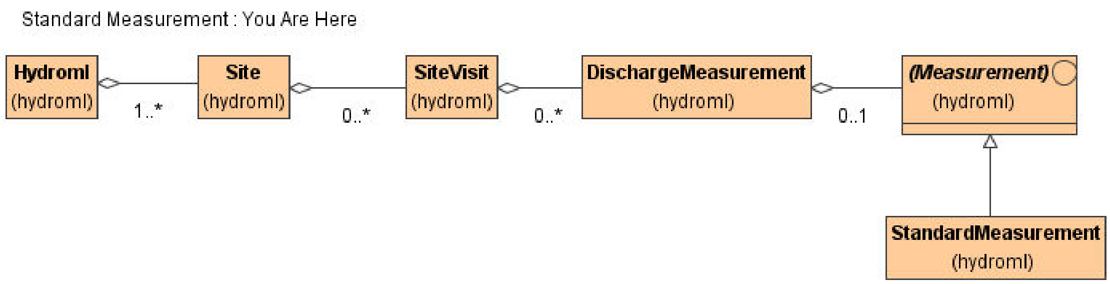

Figure 21: Context of StandardMeasurement within Hydroml Hierarchy.

Notes on StandardMeasurement:

- The VelocityMeter is the equipment used to measure velocity based on a count of revolutions.
- There is internal USGS controversy as to whether a SpinTest will be done in the field or not. We will leave it in the model.
- Some of the measurement values (areas, aggregate velocities) are computed from other measurement values. They will be included in the XML.
- Likewise, the VerticalVelocity in a section is computed from the Observations.
- Vertical Angle can occur on bridge measurements. The cable is bent back by the stream current, causing the meter to point downward as it faces upstream.
- Vertical and Horizontal angle measurements are just eyeball estimates. They should have a corresponding low degree of input precision.
- Horizontal angle is the angle of water flow to the stream perpendicular. It can be different for each section.
- The top-level velocity is the (Total Discharge / Total Cross Section Area).

Question: VerticalAngleCoeff and HorizontalAngleCoeff entities are orphan. Is this an oversight or is this from a previous design? ***This was an oversight and has been corrected.***

4.1.3 ADCP Measurement

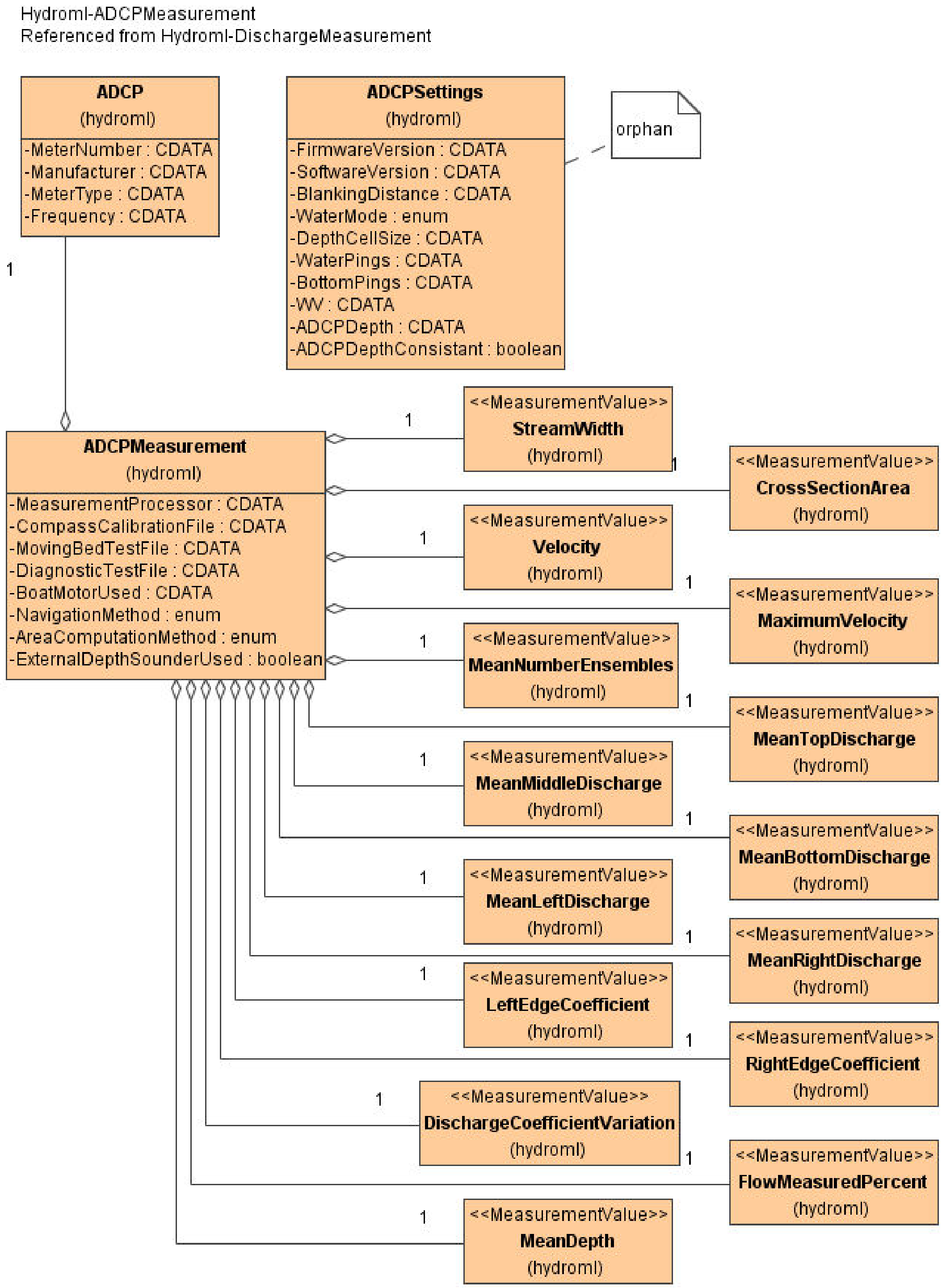

Figure 22: Hydroml-ADCPMeasurement.

ADCP is an Acoustic Doppler Profiler.

4.1.4 Volumetric Measurement

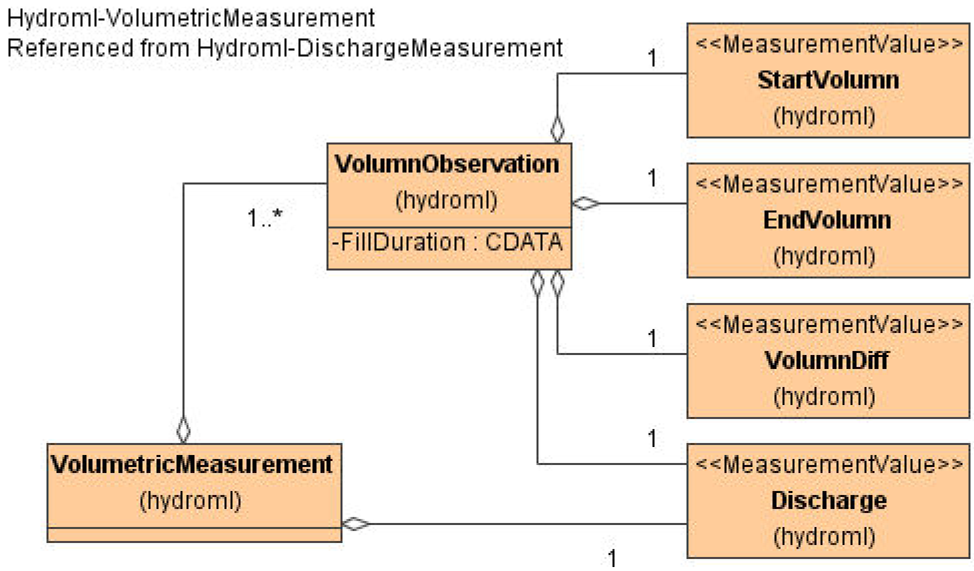

Figure 23: Hydroml-VolumetricMeasurement.

Volumetric Measurements are used for very small streams. The technician measures the time required to fill a portion of a receptacle (e.g. bucket) with a known volume.

The XML data contains start, end, and difference. Difference = end - start.

4.1.5 Flume Measurement

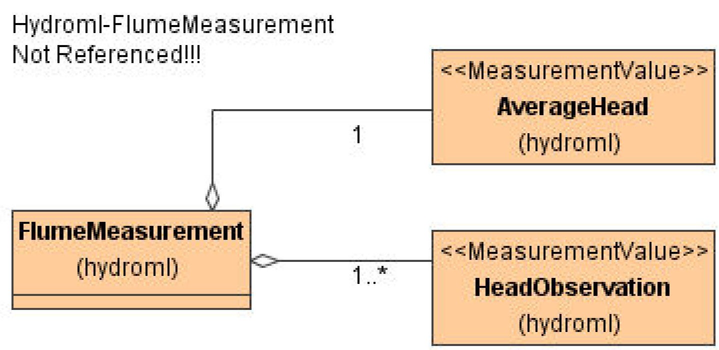

Figure 24: Hydroml-FlumeMeasurement.

Flume Measurements are also used for very small streams. A portable Flume with a known rating is inserted into the stream. The flume acts as control and contains a single cross section of a known area. The discharge is a simple relationship to the height of the water in the flume.

Recommend: In the original Hydroml, the FlumeMeasurement entity is an orphan. It needs to be listed as one of the possible measurement types contained in "DischargeMeasurement". **We agree and this has been changed.**

4.2 Control Observation

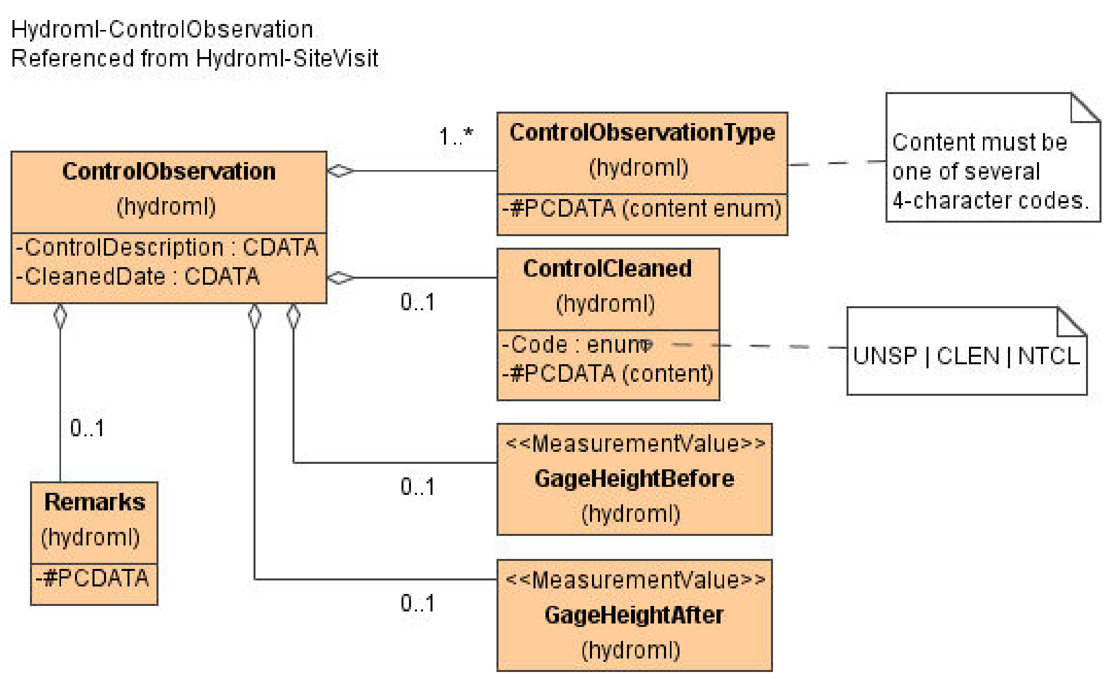

Figure 25: Hydroml-ControlObservation.

4.3 Gage Inspection

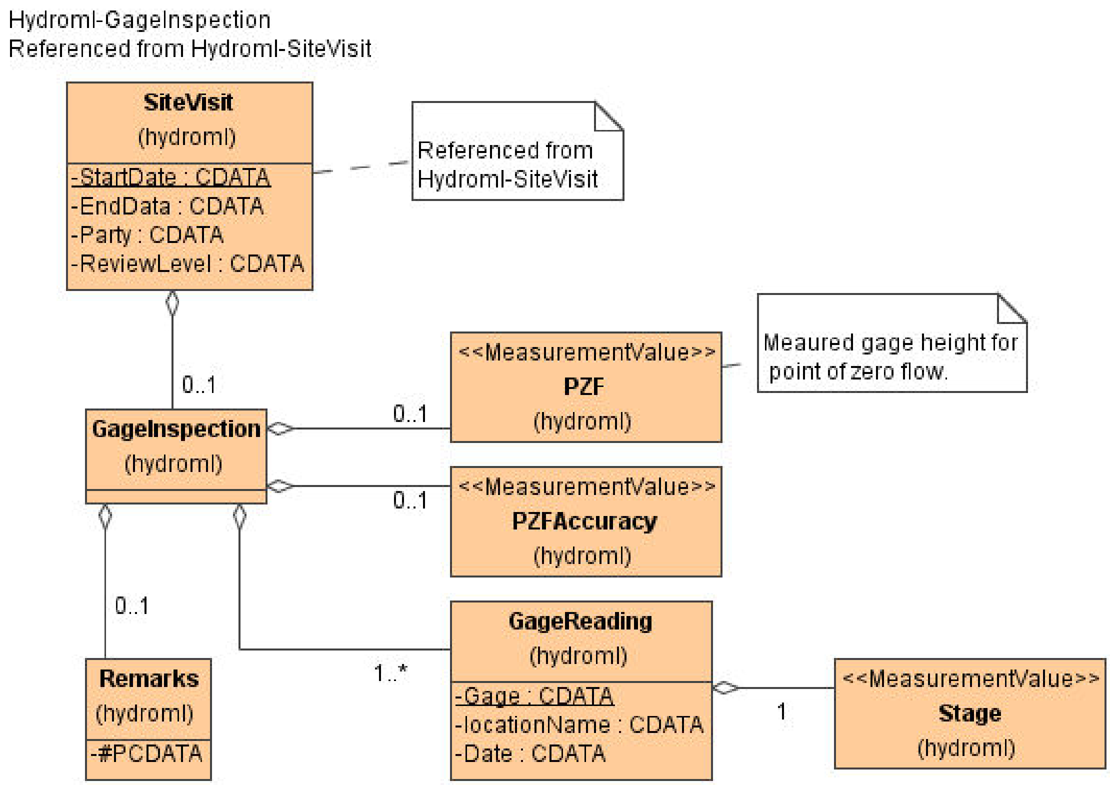

Figure 26: Hydroml-GageInspection.

4.4 Crest Stage Gage and High Water Mark Inspection

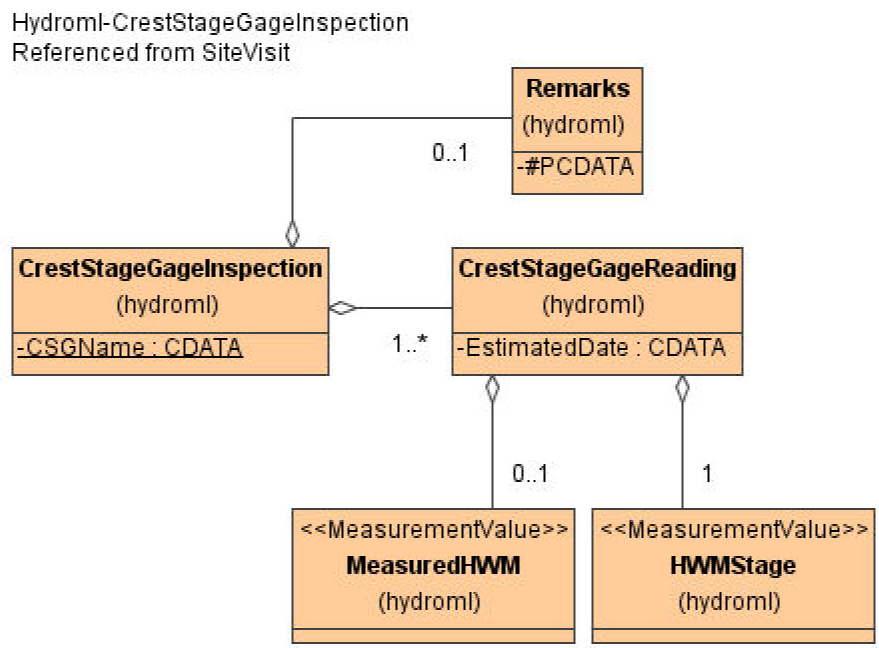

Figure 27: Hydroml-CrestStageGageInspection.

HighWaterMarkInspection is a simple entity with no subordinates. It is represented in the SiteVisit diagram.

5 Example Site Visit Hydroml File

5.1 Site Visit Hydroml File


```xml
<?xml version="1.0" standalone="no"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jeff Christman (U.S.
Geological Survey) -->
<!DOCTYPE Hydroml SYSTEM "http://water.usgs.gov/nwis_activities/xml/Hydroml.dtd">
<Hydroml UnitsFamily="English">
    <Source Agency="USGS"/>
    <Site Agency="USGS">
        <SiteName Type="USGSSiteNumber">12121212</SiteName>
        <SiteVisit StartDate="19990916151435 EST" Party="TEOLIVE">
            <DischargeMeasurement MeasurementIdentifier="88"
                    MeasurementType="WADE" MeasurementQuality="GOOD">
                <Device Type="DMX" Model="DMX-345" FirmWareVersion="R141333"
                    SoftwareVersion="v34"/>
                <Stage>1.01</Stage>
                <MeasuredDischarge DischargeCode="MEAS">150.0</MeasuredDischarge>
                <StageDifference StageDifferenceDuration="245">0.55</StageDifference>
                <StandardMeasurement NumberSections="6" SectionDescription="STEV"
                    ObservationMethod="28D" HorizontalFlowDistribution="EVEN"
                    VerticalFlowDistribution="SMTH" BaseFlow="BASE">
                    <VelocityMeter MeterType="PRAA" MeterNumber="1223222" Suspension="TROD"
                        InspectedBefore="YES" InspectedAfter="YES"/>
                    <StreamWidth>100.00</StreamWidth>
                    <CrossSectionArea>1500.00</CrossSectionArea>
                    <Velocity>0.97</Velocity>
                    <StandardSection>
                        <DistanceFromIP>0.00</DistanceFromIP>
                        <Width>2.50</Width>
                        <Depth>0.00</Depth>
                        <Area>0.00</Area>
                        <VerticalDischarge>0.00</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
                        <VerticalVelocity>0.00</VerticalVelocity>
                        <Observation Revolutions="0" Period="0.00">
                            <VelocityDepth>0.00</VelocityDepth>
                            <PointVelocity>0.000</PointVelocity>
                        </Observation>
                    </StandardSection>
                    <StandardSection>
                        <DistanceFromIP>5.00</DistanceFromIP>
                        <Width>5.00</Width>
                        <Depth>5.00</Depth>
                        <Area>25.00</Area>
                        <VerticalDischarge>6.25</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
                        <VerticalVelocity>0.25</VerticalVelocity>
                        <Observation Revolutions="5" Period="47.47">
                            <VelocityDepth>.2</VelocityDepth>
                            <PointVelocity>0.250</PointVelocity>
                        </Observation>
                        <Observation Revolutions="5" Period="47.47">
                            <VelocityDepth>.8</VelocityDepth>
                            <PointVelocity>0.250</PointVelocity>
                        </Observation>
                    </StandardSection>
                    <StandardSection>
                        <DistanceFromIP>10.00</DistanceFromIP>
                        <Width>5.00</Width>
                        <Depth>10.00</Depth>
                        <Area>50.00</Area>
                        <VerticalDischarge>50.00</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
```

```
                        <VerticalVelocity>1.00</VerticalVelocity>
                        <Observation Revolutions="18" Period="40.41">
                            <VelocityDepth>.8</VelocityDepth>
                            <PointVelocity>1.00</PointVelocity>
                        </Observation>
                    </StandardSection>
                    <StandardSection>
                        <DistanceFromIP>15.00</DistanceFromIP>
                        <Width>5.00</Width>
                        <Depth>7.00</Depth>
                        <Area>35.00</Area>
                        <VerticalDischarge>43.70</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
                        <VerticalVelocity>1.25</VerticalVelocity>
                        <Observation Revolutions="18" Period="41.41">
                            <VelocityDepth>.2</VelocityDepth>
                            <PointVelocity>1.000</PointVelocity>
                        </Observation>
                        <Observation Revolutions="27" Period="41.16">
                            <VelocityDepth>.8</VelocityDepth>
                            <PointVelocity>1.50</PointVelocity>
                        </Observation>
                    </StandardSection>
                    <StandardSection>
                        <DistanceFromIP>20.00</DistanceFromIP>
                        <Width>5.00</Width>
                        <Depth>3.00</Depth>
                        <Area>15.00</Area>
                        <VerticalDischarge>30.0</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
                        <VerticalVelocity>2.00</VerticalVelocity>
                        <Observation Revolutions="36" Period="40.04">
                            <VelocityDepth>.2</VelocityDepth>
                            <PointVelocity>2.00</PointVelocity>
                        </Observation>
                        <Observation Revolutions="36" Period="40.04">
                            <VelocityDepth>.8</VelocityDepth>
                            <PointVelocity>2.00</PointVelocity>
                        </Observation>
                    </StandardSection>
                    <StandardSection>
                        <DistanceFromIP>25.00</DistanceFromIP>
                        <Width>2.50</Width>
                        <Depth>4.0</Depth>
                        <Area>10.00</Area>
                        <VerticalDischarge>20.00</VerticalDischarge>
                        <HorizontalAngle>1.00</HorizontalAngle>
                        <VerticalAngle>1.00</VerticalAngle>
                        <VerticalVelocity>2.00</VerticalVelocity>
                        <Observation Revolutions="36" Period="39.00">
                            <VelocityDepth>.6</VelocityDepth>
                            <PointVelocity>2.00</PointVelocity>
                        </Observation>
                    </StandardSection>
                </StandardMeasurement>
            </DischargeMeasurement>
            <CrestStageGageInspection CSGName="U.S.">
                <CrestStageGageReading EstimatedDate="19990910">
                    <MeasuredHWM>1.2</MeasuredHWM>
                    <HWMStage>12.4</HWMStage>
                </CrestStageGageReading>
                <Remarks>Cork replaced</Remarks>
            </CrestStageGageInspection>
        </SiteVisit>
    </Site>
</Hydroml>
```

5.2 Gage Inspection File


```xml
<?xml version="1.0" standalone="no"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jeff Christman (U.S.
Geological Survey) -->
<!DOCTYPE Hydroml SYSTEM
"http://water.usgs.gov/nwis_activities/xml/Hydroml.dtd">
<Hydroml UnitsFamily="English">
   <Source Agency="USGS"/>
   <Site Agency="USGS">
      <SiteName Type="USGSSiteNumber">12121212</SiteName>
      <SiteVisit StartDate="19990916151435 EST" Party="TEOLIVE">
         <EnvironmentalObservation ParameterCode="00020">
            <MeasurementValue UnitsCode="DEGC">12.5</MeasurementValue>
         </EnvironmentalObservation>
         <EnvironmentalObservation ParameterCode="00010">
            <MeasurementValue UnitsCode="DEGC">10.5</MeasurementValue>
         </EnvironmentalObservation>
         <GageInspection>
            <GageReading Gage="INSD">
               <Stage>15.55</Stage>
            </GageReading>
            <GageReading Gage="OTSD">
               <Stage>15.59</Stage>
            </GageReading>
         </GageInspection>
      </SiteVisit>
   </Site>
</Hydroml>
```

5.3 Crest Stage Gage Inspection File


```
<?xml version="1.0" standalone="no"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jeff Christman (U.S.
Geological Survey) -->
<!DOCTYPE Hydroml SYSTEM
"http://water.usgs.gov/nwis_activities/xml/Hydroml.dtd">
<Hydroml UnitsFamily="English">
   <Source Agency="USGS"/>
   <Site Agency="USGS">
      <SiteName Type="USGSSiteNumber">12121212</SiteName>
      <SiteVisit StartDate="19990916151435 EST" Party="TEOLIVE">
         <EnvironmentalObservation ParameterCode="00011">
            <MeasurementValue UnitsCode="DEGC">12.5</MeasurementValue>
         </EnvironmentalObservation>
         <EnvironmentalObservation ParameterCode="00012">
            <MeasurementValue UnitsCode="DEGC">10.5</MeasurementValue>
         </EnvironmentalObservation>
         <CrestStageGageInspection CSGName="UpStream">
            <CrestStageGageReading EstimatedDate="19990910">
               <MeasuredHWM>9.56</MeasuredHWM>
               <HWMStage>19.56</HWMStage>
            </CrestStageGageReading>
            <Remarks>Added cork</Remarks>
         </CrestStageGageInspection>
         <CrestStageGageInspection CSGName="DownStream">
            <CrestStageGageReading EstimatedDate="19990910">
               <MeasuredHWM>8.14</MeasuredHWM>
               <HWMStage>18.22</HWMStage>
            </CrestStageGageReading>
            <Remarks>Added cork</Remarks>
         </CrestStageGageInspection>
      </SiteVisit>
   </Site>
</Hydroml>
```